单链表的创建

#include<stdio.h>

#include<assert.h>

#include<malloc.h>

typedef int ElemType; //或者用宏定义 #define ElemType int

typedef struct ListNode { //结点结构体

ElemType data;

ListNode\* next; //自身类型的指针类型

}ListNode;

typedef ListNode\* List; //List为"结点类型的指针变量"

//指针的类型决定了指针向前或向后一步所走过的距离

void InitList(List\*head) { //ListNode \*\*head

\*head = (ListNode\*)malloc(sizeof(ListNode)); //接下来的两行是申请一个头结点

assert(\*head != NULL);

(\*head)->next = NULL; //非常重要，头节点的指针域先指向空

}

////尾插法创建单链表（不含头节点）

//void CreatList(List\* head) {

// \*head = (ListNode\*)malloc(sizeof(ListNode)); // \*head相当于&mylist（修改\*head才算能真正修改到mylist）

// assert(\*head != NULL);

// (\*head)->data = 1; //mylist这一个结点类型的指针始终指向第一个结点，认为其指向整个链表

// (\*head)->next = NULL;

//

// ListNode\* p = \*head; //p是一个动态的指针 首先先指向第一个结点

// for (int i = 2; i <= 10; i++) {

// ListNode\* s = (ListNode\*)malloc(sizeof(ListNode));

// assert(s != NULL);

// s->data = i;

// s->next = NULL;

// p->next = s;

// p = s;

// }

//

//}

////头插法创建单链表（不含头结点）

//void CreatList(List\* head) {

// \*head = (ListNode\*)malloc(sizeof(ListNode));

// assert(\*head != NULL);

// (\*head)->data = 1; //mylist这一个结点类型的指针始终指向第一个结点(相当于头指针)，认为其指向整个链表

// (\*head)->next = NULL;

// for (int i = 2; i <= 10; i++) { //每有一个新的结点就要保证head头指针指向整个链表的头部

// ListNode\* s = (ListNode\*)malloc(sizeof(ListNode));

// assert(s != NULL);

// s->data = i;

// s->next = \*head;

// \*head=s;

// }

//}

//头插法（有头结点）head 不用一直变了

void CreatList(List\* head) {

for (int i = 1; i <= 10; i++) {

ListNode\* s = (ListNode\*)malloc(sizeof(ListNode));

assert(s != NULL);

s->data = i;

s->next = (\*head)->next;

(\*head)->next = s;

}

}

//尾插法（含头结点）

void CreatList(List\* head){

ListNode\* p = \*head;

for (int i = 1; i <= 10; i++) {

p=p->next= (ListNode\*)malloc(sizeof(ListNode));

assert(p != NULL);

p->data = i;

p->next = NULL;

}

}

void ShowList(List head) {

ListNode\* p = head->next; //匿去了头节点的信息

while (p != NULL) {

printf("%d-->", p->data);

p = p->next;

}

printf("Nul\n");

}

void main() {

List mylist; //定义链表

InitList(&mylist); //&表示取地址

CreatList(&mylist);

ShowList(mylist);

}